Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
L1	7213	last near4 access$2	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2005/04/05 16:06
L2	448	pointer with probab$8	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2005/04/05 16:06
L3	21	2 and 1	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2005/04/05 16:29
L4	3	(("6691303") or ("6766513")).PN.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/04/05 16:22
L5	3871	(707/1).CCLS.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/04/05 16:22
L6	1965	(707/2).CCLS.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/04/05 16:22
L7	4986	(707/3).CCLS.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/04/05 16:22
L8	841	(707/8).CCLS.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/04/05 16:22

L9	1130	(707/9).CCLS.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/04/05 16:22
L10	3129	(707/100).CCLS.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/04/05 16:22
L11	1694	(707/101).CCLS.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/04/05 16:23
L14	336	(718/1).CCLS.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/04/05 16:30
L15	1177	(718/100).CCLS.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/04/05 16:30
L16	1025	(718/102).CCLS.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/04/05 16:35
L17	1776	(711/154).CCLS.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/04/05 16:36
L18	571	(711/158).CCLS.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/04/05 16:36

L19	436	(711/159).CCLS.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/04/05 16:36
L20	196	(711/160).CCLS.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/04/05 16:37
L21	832	(17 or 18 or 19 or 20) and pointer	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2005/04/05 16:38
L22	953	pointer same probability	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2005/04/05 16:38
L23	8	(17 or 18 or 19 or 20) and 22	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2005/04/05 16:38

THE ACM DIGITAL LIBRARY

💥 Feedback Report a problem Satisfaction survey

Terms used **last accessed** **pointer** **probability** **threshold**

Found **11** of **342 searched out of 342.**

Sort results by: [relevance ▼]

Display results: [expanded form ▼]

💾 Save results to a Binder

⁇ Search Tips

☐ Open results in a new window

Try an Advanced Search
Try this search in The ACM Guide

Results 1 - 11 of 11

Relevance scale ☐ ☐ ☐ ☐ ☐

1 Array regrouping and structure splitting using whole-program reference affinity

Yutao Zhong, Maksim Orlovich, Xipeng Shen, Chen Ding

June 2004 **ACM SIGPLAN Notices , Proceedings of the ACM SIGPLAN 2004 conference on Programming language design and implementation**, Volume 39 Issue 6

Full text available: 📄 pdf(202.16 KB)   Additional Information: full citation, abstract, references, citings, index terms

While the memory of most machines is organized as a hierarchy, program data are laid out in a uniform address space. This paper defines a model of *reference affinity*, which measures how close a group of data are accessed together in a reference trace. It proves that the model gives a hierarchical partition of program data. At the top is the set of all data with the weakest affinity. At the bottom is each data element with the strongest affinity. Based on the theoretical model, the paper p ...

Keywords: array regrouping, program locality, program transformation, reference affinity, reuse signature, structure splitting, volume distance

2 Predicting whole-program locality through reuse distance analysis

Chen Ding, Yutao Zhong

May 2003 **ACM SIGPLAN Notices , Proceedings of the ACM SIGPLAN 2003 conference on Programming language design and implementation**, Volume 38 Issue 5

Full text available: 📄 pdf(297.60 KB)   Additional Information: full citation, abstract, references, citings, index terms

Profiling can accurately analyze program behavior for select data inputs. We show that profiling can also predict program locality for inputs other than profiled ones. Here locality is defined by the distance of data reuse. Studying whole-program data reuse may reveal global patterns not apparent in short-distance reuses or local control flow. However, the analysis must meet two requirements to be useful. The first is efficiency. It needs to analyze all accesses to all data elements in full-size ...

Keywords: data locality, pattern recognition, prediction, profiling, program locality, reuse distance, sampling, stack distance, training

3 Input/Output: Energy conservation techniques for disk array-based servers

Eduardo Pinheiro, Ricardo Bianchini

June 2004 **Proceedings of the 18th annual international conference on Supercomputing**

Full text available: 📄 pdf(174.29 KB)   Additional Information: full citation, abstract, references, citings, index terms

In this paper, we study energy conservation techniques for disk array-based network servers. First, we introduce a new conservation technique, called Popular Data Concentration (PDC), that migrates frequently accessed data to a subset of the disks. The goal is to skew the load towards a few of the disks, so that others can be transitioned to low-power modes. Next, we introduce a user-level file server that takes advantage of PDC. In the context of this server, we compare PDC to the Massive Array ...

Keywords: disk power, energy conservation, network servers

4 Using the heap to eliminate stack accesses
Zoltan Somogyi, Peter J. Stuckey
October 2002 **Proceedings of the 4th ACM SIGPLAN international conference on Principles and practice of declarative programming**

Full text available: pdf(301.52 KB)   Additional Information: full citation, abstract, references, index terms, review

The value of a variable is often given by a field of a heap cell, and frequently the program will pick up the values of several variables from different fields of the same heap cell. By keeping some of these variables out of the stack frame, and accessing them in their original locations on the heap instead, we can reduce the number of loads from and stores to the stack at the cost of introducing a smaller number of loads from the heap. We present an algorithm that finds the optimal set of varia ...

Keywords: heap cells, maximal matching, stack accesses, stack frames

5 Semantic query caching in a mobile environment
Ken. C. K. Lee, H. V. Leong, Antonio Si
April 1999 **ACM SIGMOBILE Mobile Computing and Communications Review**, Volume 3 Issue 2

Full text available: pdf(1.41 MB)   Additional Information: full citation, abstract, references, citings

Caching of remote data in a mobile client's local storage can improve data access performance and data availability. Traditional approaches are page-based, without taking advantage of the semantics of cached data. It is difficult for a client to determine if a query could be answered entirely based on locally cached data, forcing it to contact the database server for additional data. We propose a *semantic caching mechanism* which allows data to be cached as a collection of possibly related ...

6 Practical byzantine fault tolerance and proactive recovery
Miguel Castro, Barbara Liskov
November 2002 **ACM Transactions on Computer Systems (TOCS)**, Volume 20 Issue 4

Full text available: pdf(1.63 MB)   Additional Information: full citation, abstract, references, citings, index terms, review

Our growing reliance on online services accessible on the Internet demands highly available systems that provide correct service without interruptions. Software bugs, operator mistakes, and malicious attacks are a major cause of service interruptions and they can cause arbitrary behavior, that is, Byzantine faults. This article describes a new replication algorithm, BFT, that can be used to build highly available systems that tolerate Byzantine faults. BFT can be used in practice to implement re ...

Keywords: Byzantine fault tolerance, asynchronous systems, proactive recovery, state machine replication, state transfer

7 An inter-reference gap model for temporal locality in program behavior
Vidyadhar Phalke, Bhaskarpillai Gopinath
May 1995 **ACM SIGMETRICS Performance Evaluation Review , Proceedings of the 1995 ACM SIGMETRICS joint international conference on Measurement and**

modeling of computer systems, Volume 23 Issue 1

Full text available: 📄 pdf(1.16 MB)   Additional Information: <u>full citation</u>, <u>abstract</u>, <u>references</u>, <u>citings</u>, <u>index terms</u>

The property of locality in program behavior has been studied and modelled extensively because of its application to memory design, code optimization, multiprogramming etc. We propose a *k* order Markov chain based scheme to model the sequence of time intervals between successive references to the same address in memory during program execution. Each unique address in a program is modelled separately. To validate our model, which we call the Inter-Reference Gap (IRG) model, we show substant ...

Keywords: Markov chains, dynamic memory management, locality of reference, memory replacement, prediction, trace compaction, trace driven simulation

8 A caching and streaming framework for mulitmedia

Shantanu Paknikar, Mohan Kankanhalli, K. R. Ramakrishnan, S. H. Srinivasan, Lek Heng Ngoh
October 2000 **Proceedings of the eighth ACM international conference on Multimedia**

Full text available: 📄 pdf(642.08 KB)   Additional Information: <u>full citation</u>, <u>abstract</u>, <u>references</u>, <u>citings</u>, <u>index terms</u>

In this paper, we explore the convergence of the *caching* and *streaming* technologies for Internet multimedia. The paper describes a design for a streaming and caching architecture to be deployed on broadband networks. The basis of the work is the proposed Internet standard, Real Time Streaming Protocol (RTSP), likely to be the *de-facto* standard for web-based A/V caching and streaming, in the near future. The proxies are all managed by an `Intelligent Agent' or `Broker' - t ...

Keywords: broker, caching, hit ratio, layered coding, proxies, quality hit ratio, replacement policy, streaming

9 Buffer management in shared-memory Time Warp systems

Richard M. Fujimoto, Kiran S. Panesar
July 1995 **ACM SIGSIM Simulation Digest , Proceedings of the ninth workshop on Parallel and distributed simulation**, Volume 25 Issue 1

Full text available: 📄 pdf(1.39 MB) 📄   Additional Information: <u>full citation</u>, <u>abstract</u>, <u>references</u>, <u>citings</u>, <u>index terms</u>
<u>Publisher Site</u>

Mechanisms for managing message buffers in Time Warp parallel simulations executing on cache-coherent shared-memory multiprocessors are studied. Two simple buffer management strategies called the sender pool and receiver pool mechanisms are examined with respect to their efficiency, and in particular, their interaction with multiprocessor cache-coherence protocols. Measurements of implementations on a Kendall Square Research KSR-2 machine using both synthetic workloads and benchmark applica ...

Keywords: Kendall Square Research KSR-2 machine, buffer management, buffer storage, cache-coherent shared-memory multiprocessors, discrete event simulation, mall-granularity parallel simulation applications, message buffer memory, message buffers, message passing, multiprocessing programs, multiprocessor cache-coherence protocols, multiprocessor-based parallel simulators, partitioned buffer pool approach, partitioned pool mechanism, receiver pool, sender pool, severe performance degradations, shared memory systems, shared-memory time warp systems, storage management, time warp simulation

10 Decentralized storage systems: Taming aggressive replication in the Pangaea wide-area file system

Yasushi Saito, Christos Karamanolis, Magnus Karlsson, Mallik Mahalingam
December 2002 **ACM SIGOPS Operating Systems Review**, Volume 36 Issue SI

Full text available: 📄 pdf(1.93 MB)   Additional Information: <u>full citation</u>, <u>abstract</u>, <u>references</u>

Pangaea is a wide-area file system that supports data sharing among a community of widely distributed users. It is built on a symmetrically decentralized infrastructure that consists of commodity computers provided by the end users. Computers act autonomously to serve data to their local users. When possible, they exchange data with nearby peers to improve the system's overall performance, availability, and network economy. This approach is realized by aggressively creating a replica of a file w ...

11 Papers: A survey of web caching schemes for the Internet

Jia Wang

October 1999 **ACM SIGCOMM Computer Communication Review,** Volume 29 Issue 5

Full text available: pdf(1.15 MB)   Additional Information: full citation, abstract, references, citings

The World Wide Web can be considered as a large distributed information system that provides access to shared data objects. As one of the most popular applications currently running on the Internet, the World Wide Web is of an exponential growth in size, which results in network congestion and server overloading. Web caching has been recognized as one of the effective schemes to alleviate the service bottleneck and reduce the network traffic, thereby minimize the user access latency. In this pap ...

Results 1 - 11 of 11

ProQuest®

Basic Search | **Advanced Search** | **Topic Guide** | **Publication Search** | **Marked List** : 0 documents **My Research Summary**

Interface language:
English

Results

2 documents found for: *last accessed and pointer* [Set up Alert] About

(All sources) (Trade Publications) (Newspapers)

☐ Mark / Clear all on page | View marked documents   📄 Show all documents   Sort results by: Most recent first

---

☐ 1. **A Database Fix for the File System**
Jon William Toigo. **Network Computing.** Manhasset: Nov 18, 2004. Vol. 15, Iss. 23; p. 67 (3 pages)

   📄 Text+Graphics   📄 Page Image - PDF   📄 Abstract

☐ 2. **Avoiding information leaks from PC; [Computimes, 2* Edition]**
Raphael Phan Chung Wei. **New Straits Times.** Kuala Lumpur: Sep 25, 2003. p. 27

   📄 Full text   📄 Abstract

1-2 of 2

Want an alert for new results sent by email? [Set up Alert] About

Results per page: 30

Basic Search

(Tools: Search Tips   Browse Topics   1 Recent Searches)

last accessed and pointer   [Search] [Clear]

Database: Multiple databases...   ☑ Select multiple databases

Date range: All dates

Limit results to: ☑ Full text documents only 📄

              ☐ Scholarly journals, including peer-reviewed 🎓 About

More Search Options

From:ProQuest
COMPANY